Discriminative Training of the Hidden Vector State Model for Semantic Parsing

Deyu Zhou and Yulan He

Abstract—In this paper, we discuss how discriminative training can be applied to the Hidden Vector State (HVS) model in different task domains. The HVS model is a discrete Hidden Markov Model (HMM) in which each HMM state represents the state of a push-down automaton with a finite stack size. In previous applications, Maximum Likelihood estimation (MLE) is used to derive the parameters of the HVS model. However, MLE makes a number of assumptions and unfortunately some of these assumptions do not hold. Discriminative training, without making such assumptions, can improve the performance of the HVS model by discriminating the correct hypothesis from the competing hypotheses. Experiments have been conducted in two domains: the travel domain for the semantic parsing task using the DARPA Communicator data and the ATIS data, and the bioinformatics domain for the information extraction task using the GENIA corpus. The results demonstrate modest improvements of the performance of the HVS model using discriminative training. In the travel domain, discriminative training of the HVS model gives a relative error reduction rate of 31% in F-measure when compared with MLE on the DARPA Communicator data and 9% on the ATIS data. In the bioinformatics domain, a relative error reduction rate of 4% in F-measure is achieved on the GENIA corpus.

Index Terms—Semantic parsing, information extraction, hidden vector state model, discriminative training.

1 INTRODUCTION

Semantic parsing, mapping an input sentence into a structured representation of its meaning, can be applied into several applications such as spoken language understanding, information extraction etc. An example of semantic parsing for extracting protein-protein interactions is given in Figure 1. The original sentence is mapped to a semantic parse tree from which the protein-protein interactions could be easily extracted.

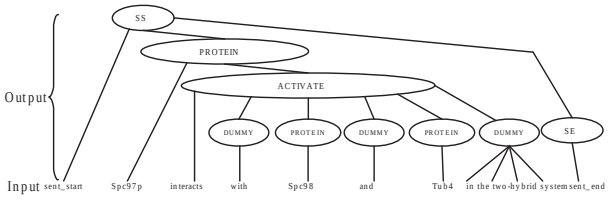


Fig. 1. An example of semantic parsing for protein-protein interactions extraction.

Traditionally, research in the field of semantic parsing can be divided into two categories: rule-based approaches and statistical approaches. Based on hand-crafted semantic grammar rules, rule-based approaches fill slots in semantic frames using word pattern and semantic tokens [1], [2]. Such rule-based approaches are typically domain-specific and often fragile. Statistical approaches are generally based on stochastic models. Given a model and an observed word sequence $W = (w_1 \cdots w_T)$, semantic parsing can be viewed as a pattern recognition problem and statistical decoding can be used to find the most likely semantic representation. If assuming that the hidden data take the form of a semantic parse tree C then the model should be a push-down automata which can generate the pair $\langle W, C \rangle$ through some canonical sequence of moves $D = (d_1 \cdots d_T)$. That is,

$$P(W, C) = \prod_{t=1}^{T} P(d_t | d_{t-1} \cdots d_1) \quad (1)$$

Decision sequences are usually steps in some top-down or bottom-up derivation of trees. For the general case of an unconstrained hierarchical model, D will consist of three types of probabilistic move:

1) popping semantic category labels off the stack;

• *D. Zhou and Y. He are with the Informatics Research Centre, the University of Reading, Reading RG6 6BX, UK. E-mail: {d.zhou, y.he}@reading.ac.uk*

Manuscript received April 19, 2007; revised January 11, 2008.

2) pushing one or more non-terminal semantic category label onto the stack;
3) generating the next word.

In practice, conditional independence can be used to reduce the number of parameters needed to manageable proportions. As in conventional statistical language modeling, this involves defining an equivalence function Φ which groups move sequences into equivalent classes. Thus, the final generic parsing model is:

$$P(W,C) = \prod_{t=1}^{T} P(d_t|\Phi(d_{t-1}\cdots d_1)) \qquad (2)$$

The above is essentially the history-based model [3], [4] where the probability of each parser action is conditioned on the history of previous actions in the parse or some partially built structure.

Traditionally, the parsing models have been trained to have the maximum likelihood $P(W,C)$. However the goal of training the parsing models is to find the correct semantic representation C, i.e. minimize the recognition error. Although a maximum probability is often correlated with a better recognition performance, the correlation is not perfect and is not proved. We can provide some examples with high likelihood but without better recognition rate. Furthermore, MLE makes some assumptions on the model, such as the model correctly represents the underlying stochastic process, the amount of training data is infinite, and the true global maximum of the likelihood can be found. When assumptions made about the model are incorrect and the training data are not sufficient, MLE yields a suboptimal solution.

Previous research has shown that the performance of the HMM trained using MLE can often be improved further using discriminative training. Discriminative training methods based on different criterions such Maximum Mutual Information (MMI), minimum classification error (MCE) etc, have been tried. In particular, MMI estimation has been studied for speech recognition and substantial gains in performance have been reported [5] while discriminative training based on MCE [6]–[9] has also been applied for speech recognition.

An early example of a purely statistical approach to semantic parsing is the finite state semantic tagger used in AT&T's CHRONUS system [10]. In this system, utterance generation is modeled by an HMM-like process in which the hidden states correspond to semantic concepts and the state outputs correspond to the individual words. This model is sometimes referred to as the flat-concept model to emphasize its inability to represent hierarchical structure. For the constrained case of the flat concept model, the stack is effectively depth one and $\langle W,C\rangle$ is built by repeatedly popping one label off the stack, pushing one new label onto the stack and then generating the next word. This kind of model is unable to capture long distance dependencies. This inability of representing hierarchical structures can be overcome by allowing the state stack to grow without limit and more than one new semantic labels to be pushed onto the stack. This is essentially analogous to using stochastic phrase structure rules and extends the class of supported languages from *regular* to *context-free*. The Hidden Understanding Model (HUM) model [11]–[14] is an early example of such an SCFG model which uses fully-annotated corpora to simplify the parameter estimation problem that is otherwise complex due to the recursive nature of hierarchical parse trees.

A general SCFG model is computationally expensive to train. However, computational tractability issues may be tackled by imposing certain constraints on the SCFG model itself. The hierarchical hidden Markov model (HHMM) model [15] constrains the level of hierarchies or the state stack depth to be a bounded depth, it is nevertheless still complex as its state inference takes $O(T^3)$ time where T is the sequence length. Murphy converts an HHMM into a dynamic Bayesian networks [16] such that the inference can be done using the junction tree algorithm which only takes $O(T)$ time empirically provided that the HHMM hierarchy depth and the number of states at each level of hierarchy are bounded to some relatively small values.

The weakness of non-lexicalized SCFG models such as HUM and HHMM can be avoided by associating a headword to each non-terminal in the parse tree. Examples of lexicalized SCFG models such as the immediate-head parsing model [17] achieved 6% reduction in recall error and 5% reduction in precision error compared to a general non-lexicalized model when tested on Penn WSJ treebank data [18].

Chelba's structured language model (SLM) [19] does not impose a constraint on the state stack depth, but it does constrain the pushing of at most one new tag (a POStag in this case, not a semantic tag) into the stack. As opposed to the conventional SCFG models where each parser action is only

conditioned on the immediately preceding non-terminal tag being expanded, a parser action in the SLM is conditioned on the previously two exposed headwords. However, Chelba's SLM has the limitation that it is not able to capture dependencies between non-headwords due to its headword percolation rules. For example, *less* and *than* as in *less people join the society this month than last month* where neither *less* nor *than* are headwords of this phrase.

The flat-concept model is simple and robust to estimate. However, it can not represent nested structured information. On the other hand, the hierarchical structured models are able to capture long distance dependencies, but require fully annotated treebank data for training which are difficult to obtain in practice. A Hidden Vector State (HVS) model [20] has been proposed which extends the flat-concept HMM model by expanding each state to encode the stack of a push-down automaton. This allows the model to efficiently encode hierarchical context. At the same time, such a model can be trained using only lightly annotated data.

In this paper, we propose a discriminative approach based on parse error measure to train the HVS model. To adjust the HVS model to achieve minimum parse error, the generalized probabilistic descent (GPD) algorithm [21] was used. Experiments have been conducted in two domains: the travel domain for the semantic parsing task using the DARPA Communicator data and the ATIS data, and the bioinformatics domain for the information extraction tasks using the GENIA corpus. The results demonstrate modest improvements of the performance of the HVS model using discriminative training. In the travel domain, discriminative training of the HVS model gives a relative error reduction rate of 31% in F-measure when compared with MLE on the DARPA Communicator data and 9% on the ATIS data. In the bioinformatics domain, a relative error reduction rate of 4% in F-measure is achieved on the GENIA corpus.

The rest of the paper is organized as follows: Section 2 surveys related work. In Section 3, we briefly describe the HVS model and how it can be trained in a discriminative way. Experimental setup is discussed in Section 4 and experimental results are presented in Section 5. Finally, Section 6 concludes the paper and gives future directions.

2 Related Work

Discriminative training was initially proposed as an alternative training technique for the speech recognition problem. Historically, the predominant training technique has been maximum likelihood estimation (MLE). However, it turns out that MLE gives optimal estimates only if three conditions are satisfied.

- the model correctly represents the stochastic process;
- an infinite amount of training data are available;
- the true global maximum of the likelihood can be found.

In practice, none of the above conditions is satisfied. This is the motivation for discriminative training. Discriminative training attempts to optimize the correctness of a model by formulating an objective function that in some way penalizes parameter sets that are liable to confuse correct and incorrect answers. Many discriminative training schemes have been proposed based on different objective functions such as maximum mutual information (MMI), minimum word error (MWE), minimum phone error (MPE), minimum classification error (MCE) etc.

From an information theoretic standpoint, MMI which is shared between X and Y is the reduction of X's uncertainty due to the knowledge of Y. Given the observation O, the speech recognizer should choose a word sequence W to make sure that the correct answer has the minimal amount of uncertainty. The IBM speech recognition group was the first to report results with MMI estimation [22]. They obtained 18% lower recognition error rate in a speaker-dependent isolated word recognition system using gradient descent. After that, improvements were reported in [23] using MMI estimation for isolated word recognition. Since gradient descent for MMI estimation does not guarantee convergence and is computationally expensive, an alternative strategy is to use the extended Baum-Welch (EBW) [24]. In [5], a reduction of string error rate by close to 50% was reported using EBW on the TI/NIST connected digit database. Later, lattice-based discriminative training was proposed to optimize the parameters of a continuous density HMM-based large vocabulary recognition system using MMI criterion [25].

The minimum classification error (MCE) objective function is designed to directly minimize the errors made by the recognizer on the training set. In [9], experiments were conducted on several key speech recognition tasks and the MCE method provided a significant reduction of recognition er-

ror rate. In [26], the minimum word error (MWE) and minimum phone error (MPE) objective functions were proposed. The MWE objective function attempts to minimize the number of word level errors. Instead of maximizing the word accuracy in the MWE approach, the MPE approach is to maximize the phone level accuracy.

All the discriminative methods described above were applied in the speech recognition domain. However, there has been little work in extending them to semantic parsing. To the best of our knowledge, [27] is the only work to estimate the probabilities for a neural network statistical parser using discriminative training criterion. Experiments were conducted to compare the performance of three statistical parsers, one generative, one discriminative, one generative but using discriminative training criterion. Results were showed that the last parser outperforms the previous two and achieves 90.1% in F-measure on the Penn Treebank data.

In this paper, we propose an discriminative training approach based on minimum parse error for the HVS model. Here, minimum parse error is similar to MCE, and is used to describe the error of semantic parsing on training set. To adjust the HVS model parameters to achieve minimum parse error, the generalized probabilistic descent (GPD) algorithm [21] is used to minimize a smoothed function of parsing error along the steepest direction.

3 METHODOLOGIES

3.1 Hidden Vector State Model

All the parser models described in Section 1 apply constraints in one way or another to the general framework described in Equation 2. In particular, when considering a constrained form of automata where the stack is finite depth and $\langle W, C\rangle$ is built by repeatedly popping 0 to n labels off the stack, pushing exactly one new label onto the stack and then generating the next word, it defines the Hidden Vector State (HVS) model in which conventional grammar rules are replaced by three probability tables.

Given a word sequence W, concept vector sequence $\mathbf{C}$ and a sequence of stack pop operations N, the joint probability of $P(W, \mathbf{C}, N)$ can be decomposed as

$$P(W, \mathbf{C}, N) = \prod_{t=1}^{T} P(n_t|\mathbf{c}_{t-1}) \cdot P(c_t[1]|c_t[2 \cdots D_t]) \cdot P(w_t|\mathbf{c}_t) \quad (3)$$

where $\mathbf{c}_t$, the vector state at word position t, is a vector of D_t semantic concept labels (tags), i.e. $\mathbf{c}_t = [c_t[1], c_t[2], ..c_t[D_t]]$ where $c_t[1]$ is the preterminal concept label and $c_t[D_t]$ is the root concept label, n_t is the vector stack shift operation at word position t and take values in the range $0, \ldots, D_{t-1}$ and $c_t[1] = c_{w_t}$ is the new preterminal semantic tag assigned to word w_t at word position t.

Thus, the HVS model consists of three types of probabilistic move, each move being determined by a discrete probability table:

1) popping semantic labels off the stack - $P(n|\mathbf{c})$;
2) pushing a pre-terminal semantic label onto the stack - $P(c[1]|c[2 \cdots D])$;
3) generating the next word - $P(w|\mathbf{c})$.

This constrained form of automata implements a right-branching parser which has some very convenient properties. It is left-to-right, and it has complexity $O(TQ^D)$ [1], yet it can still model hierarchical structure.

3.2 Maximum Likelihood Training of the HVS Model

In the HVS-based semantic parser, the purpose of training is to find the HVS parameter set $\lambda = \{\mathbf{C}, N\}$ which will result in the decoder with the lowest possible recognition error rate. This is done by maximizing some objective function $R(\lambda)$. By far the most commonly used parameter estimation technique is Maximum Likelihood Estimation (MLE). The objective function typically used in MLE, given the observations $W = \{W_1, W_2, \ldots, W_I\}$, is

$$R(\lambda) = f_{ML}(\lambda) = \log \prod_{r=1}^{I} P(W_r, \lambda) = \sum_{r=1}^{I} \log P(W_r, \lambda) \quad (4)$$

MLE attempts to maximize the likelihood of the training data. Thus, we need to compute the λ that best explain the data, i.e.

$$\lambda^* = \underset{\lambda}{\operatorname{argmax}} \sum_{r=1}^{I} \log P(W_r, \lambda) \quad (5)$$

The most obvious quality of MLE is the existence of a re-estimation formula $f(\cdot)$ such that, if $\hat{\lambda} = f(\lambda)$, then we will have $R(\hat{\lambda}) \geq R(\lambda)$, with equality only when λ is a local maximum (or, possibly, a saddle point) of $R(\lambda)$. It can be quickly

1. Where T is the length of the sequence, D is the maximum stack depth, and Q is the maximum number of concepts (node labels) at each level of the stack.

trained using the globally convergent Baum-Welch algorithm [28].

The re-estimation formulae derived are [20]:

$$P^*(n|c') = \frac{\sum_t P(n_t=n,\mathbf{c}_{t-1}=c'|W,\lambda^k)}{\sum_t P(\mathbf{c}_{t-1}=c',W|\lambda^k)} \quad (6)$$

$$P^*(c[1]|c[2..D]) = \frac{\sum_t P(\mathbf{c}_t,W|\lambda^k)}{\sum_t P(c_t[2..D]=c[2..D]|W,\lambda^k)} \quad (7)$$

$$P^*(w|\mathbf{c}) = \frac{\sum_t P(\mathbf{c}_t=\mathbf{c},w_t=w|\lambda^k)}{\sum_t P(\mathbf{c}_t=\mathbf{c},W|\lambda^k)} \quad (8)$$

MLE makes a number of assumptions: observation are from a known family of distribution, training data are unlimited, and the global maximum of the likelihood can be found. Unfortunately, in general none of these assumptions holds. Given that MLE's assumptions are in general not satisfied, it is not guaranteed to produce optimal results. Also MLE is sub-optimal as it only aims to maximize the correct model and ignores the impact of other incorrect competing models. This has led researchers to explore the feasibility and efficiency of using discriminative training.

3.3 Discriminative Training of the HVS Model

Normally, MLE is used for generative statistical model training in which only the correct model needs to be updated during training. It is believed that improvement can be achieved by training the generative model based on a discriminative optimization criterion [29] in which the training procedure is designed to maximize the conditional probability of the parses given the sentences in the training corpus. That is, not only the likelihood for the correct model should be increased but also the likelihood for the incorrect models should be decreased.

Given a word sequence W, a semantic parser needs to compute the most likely set of embedded concepts $\hat{C}$ by maximizing the following equation:

$$P(C|W) = \frac{P(W|C)P(C)}{P(W)} \quad (9)$$

For a given W, $P(W)$ is a constant and therefore

$$\hat{C} = \underset{C}{\operatorname{argmax}}\ P(W|C)P(C) \quad (10)$$

where $P(W|C)$ is called the lexical model and $P(C)$ is called the semantic model.

Traditionally, semantic parsing models have been trained to have the maximum likelihood $P(C)$. Although, a higher $P(C)$ is often related with a better performance, the correlation is not perfect. Discriminative training based on minimum parse error is therefore proposed here.

Assuming the most likely semantic parse tree is $\hat{C} = C_j$ and there are altogether M semantic parse hypotheses for a particular sentence W, a parse error measure [6]–[8] can be defined as

$$d(W) = -\log P(W,C_j) + \log[\frac{1}{M-1}\sum_{i,i\neq j} P(W,C_i)^\eta]^{\frac{1}{\eta}} \quad (11)$$

where η is a positive number and is used to select competing semantic parses. When $\eta = 1$, the competing semantic parse term is the average of all the competing semantic parse scores. When $\eta \to \infty$, the competing semantic parse term becomes $\max_{i.i\neq j} P(W,C_i)$ which is the score for the top competing semantic parse result. By varying the value of η, we can take all the competing semantic parses into consideration. $d(W) > 0$ implies classification error and $d(W) \leq 0$ implies correct decision.

The sigmoid function can be used to normalize $d(W)$ in a smooth zero-one range and the loss function is thus defined as [6]:

$$\ell(W) = \text{sigmoid}(d(W)) \quad (12)$$

where

$$\text{sigmoid}(x) = \frac{1}{1+e^{-\gamma x}} \quad (13)$$

Here, γ is a constant which controls the slope of the sigmoid function.

For a given training data set consisting of I samples $\{W_1,\ldots,W_I\}$, the empirical probability measure P_I defined on the training data set is a discrete probability measure that assigns equal mass at each sample. The empirical loss, on the other hand, is thus expressed as

$$L_0(\lambda) = \frac{1}{I}\sum_{j=1}^{I}\sum_{i=1}^{M}\ell_i(W_j,\lambda) = \int \ell(W,\lambda)dP_I \quad (14)$$

The expected loss is defined as

$$L(\lambda) = E_W\{\ell(W,\lambda)\} \quad (15)$$

It has been shown that the empirical loss defined on the I independent training samples will converge to the expected loss, as the sample size I increases.

The update formula is given by:

$$\lambda^{k+1} = \lambda^k - \epsilon^k \nabla\ell(W_i,\lambda^k) \quad (16)$$

where ϵ^k is the step size.

Using the definition of $\ell(W_i, \lambda^k)$ and after working out the mathematics, the following update formulae can be obtained:

$$(\log P(n|\mathbf{c}'))^* = \log P(n|\mathbf{c}') - \epsilon\gamma\ell(d_i)(1-\ell(d_i))\times [-I(C_j, n, \mathbf{c}') + \sum_{i,i\neq j} I(C_i, n, \mathbf{c}')\frac{P(W_i, C_i, \lambda)^\eta}{\sum_{i,i\neq j} P(W_i, C_i, \lambda)^\eta}] \quad (17)$$

$$(\log P(c[1]|c[2..D]))^* = \log P(c[1]|c[2..D]) - \epsilon\gamma\ell(d_i)(1-\ell(d_i)) \times [-I(C_j, c[1], c[2..D]) + \sum_{i,i\neq j} I(C_i, c[1], c[2..D])\frac{P(W_i, C_i, \lambda)^\eta}{\sum_{i,i\neq j} P(W_i, C_i, \lambda)^\eta}] \quad (18)$$

$$(\log P(w|\mathbf{c}))^* = \log P(w|\mathbf{c}) - \epsilon\gamma\ell(d_i)(1-\ell(d_i))\times [-I(C_j, w, \mathbf{c}) + \sum_{i,i\neq j} I(C_i, w, \mathbf{c})\frac{P(W_i, C_i, \lambda)^\eta}{\sum_{i,i\neq j} P(W_i, C_i, \lambda)^\eta}] \quad (19)$$

where $I(C_i, n, \mathbf{c}')$ denotes the number of times the operation of popping up n semantic tags at the current vector state $\mathbf{c}'$ in the C_i parse tree, $I(C_i, c[1], c[2..D])$ denotes the number of times the operation of pushing the semantic tag $c[1]$ at the current vector state $c[2..D]$ in the C_i parse tree and $I(C_i, w, \mathbf{c})$ denotes the number of times of emitting the word w at the state $\mathbf{c}$ in the parse tree C_i.

A full derivation of the update formulae is given in the Appendix.

3.4 Framework of Discriminative Training

Figure 2 shows the overall discriminative training procedure for the HVS model. The model is originally trained by the MLE criteria. The MLE-trained model is then used to parse the sentences from the lightly annotated training corpus. For each training sentence, the parse results are output as a parse lattice. An example is shown in Figure 3 where the correct parse path is highlighted with the bold line. For each individual word, the count relating to the correct parse decision is increased while the count to the incorrect parse decisions is decreased[2] . Thus, the model is trained to separate the correct parse from those incorrect parses. The discriminatively trained model is then used to parse the training sentences again and the whole training procedure repeats until no significant error reduction is observed in the held-out set.

2. In our implementation, we use the top N competing parses instead of all the incorrect parses presented in the lattice.

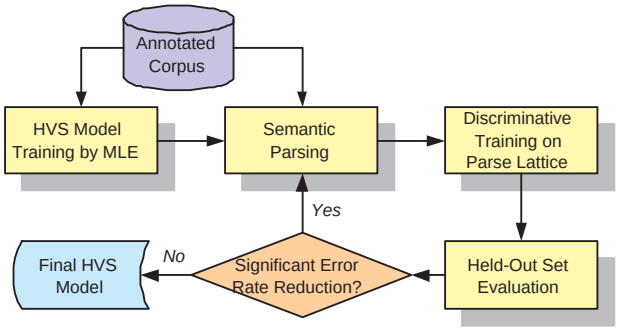


Fig. 2. The discriminative training procedure.

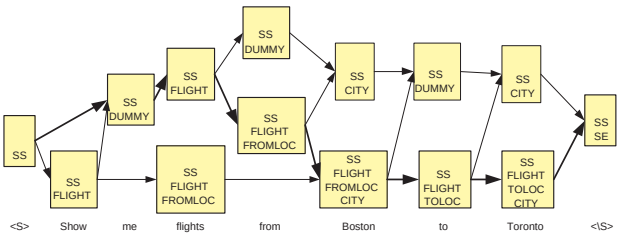


Fig. 3. An example of semantic parse lattice.

TABLE 1
Statistics of the three corpora used.

Dataset	Training set	Testing set
DARPA Communicator	12702	1178
ATIS	4978	893
GENIA corpus	1600	250

4 Experimental Setup

Experiments have been conducted on the three corpora in the two domains: the DARPA Communicator data and the ATIS data in the travel domain, the GENIA corpus in the bioinformatics domain. Table 1 gives the overall statistics of these three corpora. The following describes experimental setup as well as the evaluation matric used in the experiments.

4.1 DARPA Communicator Data

The DARPA Communicator data [30] are available to the public as open source download. The data contain utterance transcriptions and the semantic parse results from the rule-based Phoenix parser [3]. The DARPA Communicator data were collected in 461 days and consist of 2211 dialogues or 38408 utterances in total. From these, 46 days were randomly selected for use as test set data and the remainder were used for training. After cleaning

3. http://communicator.colorado.edu/phoenix

up the data, the training data consist of 12702 utterances while the test set contains 1178 utterances. Since in our discriminative training framework the held-out set is needed, we split the test set into two parts with the same size, one is used as the held-out set and the other is used to evaluate the performance of discriminative training.

The abstract annotation used for training and the reference annotation needed for testing were derived by hand correcting the Phoenix parse results. An example of a reference frame is:

Show me flights from Boston to New York.	
Frame:	FLIGHT
Slots:	FROMLOC.CITY = Boston
	TOLOC.CITY = New York

Performance was then measured in terms of F-measure on slot/value pairs, which combines the precision and recall with equal weight and is defined as $2\times\text{Recall}\times\text{Precision}/(\text{Recall+Precision})$. Recall measures how much relevant information the method has extracted. It is defined as the percentage of correct answers given by the method over the total actual correct answers. Precision measures how much of the information the system extracted is correct. It is defined as the percentage of correct answers given by the method over all the answers extracted by the method.

4.2 ATIS

The Air Travel Information Services (ATIS) corpus [31] contains air travel information data. The ATIS training set consists of 4978 utterances selected from the Class A (context independent) training data in the ATIS-2 and ATIS-3 corpora while the ATIS test set contains both the ATIS-3 NOV93 and DEC94 datasets. Abstract semantics for each training utterance were derived semi-automatically from the SQL queries provided in ATIS-3. After the parse results have been generated for the test sets, post-processing is performed to extract relevant slot/value pairs and convert them into a format compatible with the reference frames.

4.3 GENIA Corpus

Protein-protein interactions (PPIs) referring to the associations of protein molecules are crucial for many biological functions. A major challenge in text mining for biomedicine is automatically extracting protein-protein interactions from the vast amount of biomedical literature since most knowledge about them still hides in biomedical publications. We have constructed an information extraction system based on a semantic parser employing the HVS model for PPIs [32].

GENIA [33] is a collection of 2000 research abstracts selected from the search results of MEDLINE database using keywords (MESH terms) *"human, blood cells and transcription factors"*. All these abstracts were then split into sentences and those containing more than two protein names and at least one interaction keyword were kept. Altogether 3533 sentences were left and 2500 sentences were sampled to build our data set.

Abstract annotation were derived manually. An example of such an annotation together with the PPI information embedded is:

Sentences:	CUL-1 was found to interact with SKR-1, SKR-2, SKR-3, SKR-7, SKR-8 and SKR-10 in yeast two-hybrid system.
Annotation:	PROTEIN_NAME(INTERACT (PROTEIN_NAME))
PPI :	CUL-1 interact SKR-1
	CUL-1 interact SKR-2
	CUL-1 interact SKR-3
	CUL-1 interact SKR-7
	CUL-1 interact SKR-8
	CUL-1 interact SKR-10

The evaluation of the experimental results is based on the values of TP (true positive), FP (false positive), and FN (false negative). TP is the number of correctly extracted interactions. (TP+FN) is the number of all interactions in the test set and (TP+FP) is the number of all extracted interactions.

F-measure is computed using the formula below:

$$\text{F-measure} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \qquad (20)$$

where Recall is defined as TP/(TP+FN) and Precision is defined as TP/(TP+FP).

5 Experimental Results

This section presents the evaluation results in details.

5.1 Results based on MLE Training of the HVS Model

Firstly, experiments were conducted to find the smoothing technique which yielded the best result. For the data in the bioinformatics domain,

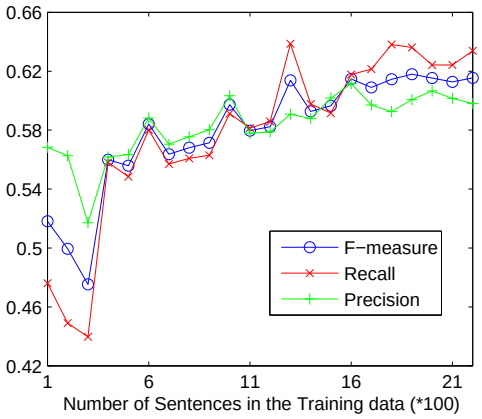


Fig. 4. Performances of the HVS model trained on the increasingly added training data.

experiments were further conducted to find the proper size of the training data to generate the best performance since in the GENIA corpus, the size of the training set is not predefined and it is crucial for statistical model training.

5.1.1 Smoothing Techniques

The performances of the HVS models on the three corpora using different smoothing techniques are listed in Table 2. It can be observed that the best performance was achieved using Witten-Bell for both the stack shift operation and output probabilities.

5.1.2 GENIA Corpus

To explore the best performance of the HVS model on the GENIA corpus, we need to determinate the proper size of the training data. Experiments have been conducted as follows. The corpus was first randomly split into the training set and the test set at the ratio of 9:1. The test set consists of 250 sentences and the remaining 2250 sentences were used as the training set. The split were conducted ten times with different training and test data each round. For each split, 100 sentences were randomly selected from the training set to build an initial HVS model which was then tested on the test set. Then another 100 sentences were added from the training set to build a new HVS model and its performance was analyzed again. This procedure was repeated until all the 2250 sentences were added into the training set. Figure 4 illustrates the performance at each stage.

It shows that the model performance gradually improves when adding more training data. It saturates when the size of the training data reaches 1600. At this point, the average F-measure value obtained is 61.47% with the balanced recall and precision values. This implies that for the GENIA corpus, 1600 sentences would be sufficient to train the HVS model. Thus, the training set size of 1600 is fixed for discriminative training in the subsequent experiments.

5.2 Results based on Discriminative Training of the HVS Model

Experiments have been conducted on the DARPA Communicator data, the ATIS data and the GENIA corpus by discriminatively training the HVS model based on the update formulae (17)-(19). The following parameters were used: $\gamma = 0.5$, $\eta = 0.1$ and $\epsilon = 0.5$.

5.2.1 Optimal Size of Training Data

The size of the training set for discriminative training is highly correlated with the performance of the resulted HVS model. In our experiments, the size of training set is determined by the parameters: N, the number of semantic parse hypotheses and I, the number of utterances in the training data. To reveal the relationship between the performance of discriminative training and the size of the training data, experiments were conducted in the following way on the DARPA Communicator data:

- Set I = 12702 (the size of the whole training data), 5000 (almost half of the size of the whole training data), 1000, 500, 200, 100.
- After fixing the value of I, randomly sampled I utterances from the whole training set (12702 utterances) in 10 times. At each time τ, a training set S_τ with size of I was constructed. We only sampled once for I = 12702, and sampled ten times for all the other values of I.
- For each training set S_τ, $\tau = 1, ...10$, discriminative training was conducted when setting N = 5, 10, 20, 30, 40.

Table 3 lists the best performance among the experiments for various I and N on the DARPA Communicator data. It can be observed that the best performance of the HVS model using discriminative training is achieved when $I = 1000$ and $N = 30$. It should be noted that a filtering method has been employed to construct the training set by selecting sentences with semantic parse probabilities exceeding certain threshold. This is to reduce the possible errors introduced to discriminative training since only abstract annotation is

TABLE 2
Performance comparison of MLE training of the HVS model using various smoothing techniques.

Smoothing methods		ATIS			DARPA Communicator			GENIA		
Stack Shift Operation	Output Probability	Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
Linear	Nonlinear	86.95%	91.9%	89.35%	83.35%	85.93%	84.62%	58.56%	58.14%	58.35%
	Linear	86.66%	91.53%	89.03%	83.94%	85.97%	84.94%	57.86%	57.58%	57.72%
	Witten-Bell	90.14%	91.97%	90.93%	87.58%	87.54%	87.56%	61.21%	60.52%	60.87%
NonLinear	NonLinear	86.24%	91.22%	88.66%	83.35%	85.77%	84.54%	56.49%	57.81%	57.14%
	Linear	86.09%	91.07%	88.51%	83.89%	86.33%	85.09%	58.09%	60.33%	59.19%
	Witten-Bell	89.61%	91.82%	90.7 %	87.22%	87.02%	87.12%	59.18%	60.47%	59.82%
Witten-Bell	NonLinear	87.02%	92.22%	89.54%	82.94%	86.11%	84.50%	61.86%	57.22%	59.45%
	Linear	86.91%	92.14%	89.45%	83.26%	86.40%	84.80%	60.75%	58.27%	59.48%
	Witten-Bell	**90.21%**	**92.04%**	**91.11%**	**87.81%**	**88.13%**	**87.97%**	**61.78%**	**61.16%**	**61.47%**

provided for each sentence instead of the word-level annotation or the full semantic parse path.

TABLE 3
The best performance of 10 times sampling experiments on various I and N on the DARPA Communicator data.

N \ I	100	200	500	1000	5000	12702
5	91.39%	91.45%	91.1%	91.15%	83.70%	75.60%
10	91.01%	91.29%	91.37%	91.31%	90.71%	76.28%
20	91.13%	90.87%	91.38%	91.55%	91.58%	84.62%
30	90.99%	90.9%	91.3%	**91.68**%	91.56%	90.74%
40	90.9%	91.1%	91.15%	91.57%	91.64%	91.19%

To examine the performance of the HVS model in each sampling in more details, Figure 5 gives the boxplot of the performance of the HVS model in each sampling, showing the variation of the performance of the resulted HVS model as a function of N and I on the DARPA Communicator data. It shows that when $I = 1000$ and $N = 30$, this size of the training set gives the best and balanced performance among all the candidate training sets.

For the AITS data, experiments were conducted in a similar way. Table 4 lists the best performance among the experiments for various I and N on the ATIS data. It can be observed that the best performance using discriminative training is achieved when $I = 100$ and $N = 5$ or 10.

For the GENIA corpus, the whole training data (1600 utterances) were split into 8 non-overlapping sets with each set consisting of 200 sentences. The training set size 200 was chosen empirically as the performance of the HVS model would degrade if setting the training set size to 100 or 500.

To explore the convergence rate of discriminative training, we further analyzed the experiments on the DARPA Communicator data. Figure 6 gives the histogram of the iteration numbers on the experiments we have conducted to find the optimal size of the training data. As described above, overall experiments were conducted $4 \times 5 \times 10 + 5 \times 2 + 5 = 215$ times. From Figure 6, we can see that almost half of the experiments converged before iteration 3. This shows the fast convergence rate of the discriminative training method.

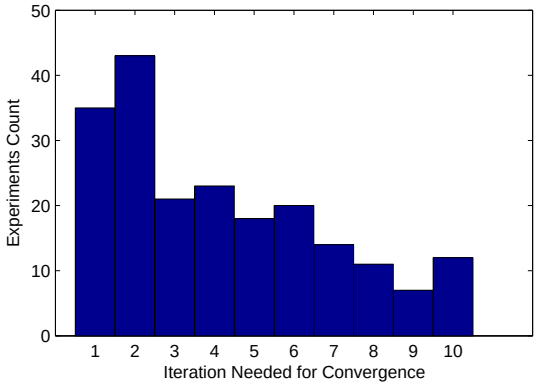


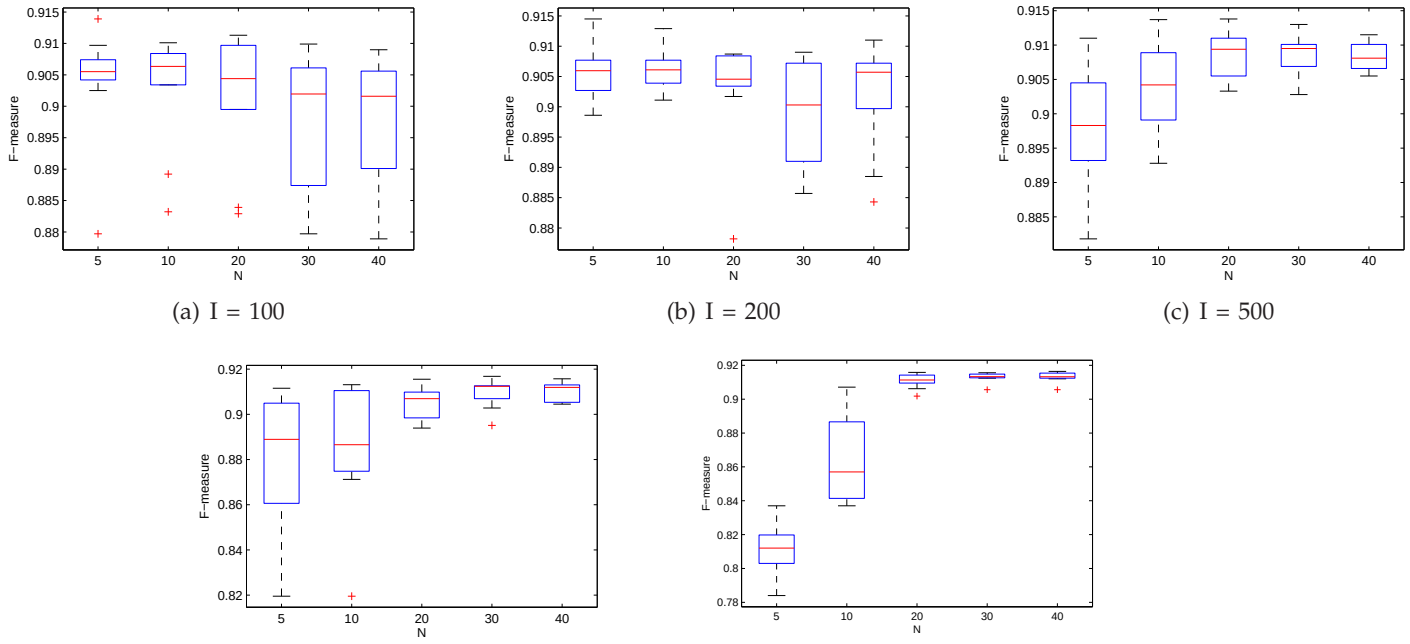
Fig. 6. Histogram of the number of experiments vs the iterations required for convergence on the DARPA Communicator data.

TABLE 4
The best performance of 10 times sampling experiments on various I and N on the ATIS data.

N \ I	100	200	500	1000	4978
5	**91.87**%	91.77%	91.78%	91.71%	87.94%
10	**91.87**%	91.82%	91.76%	91.71%	88.77%
20	91.84%	91.8%	91.72%	91.73%	89.29%

5.3 Comparison of Discriminative Training with MLE

The results using MLE and discriminative training are listed in Table 5. For the DARPA Communicator data, N and I were set to 30 and 1000 respectively, and the discriminatively trained HVS model outperforms the ML trained HVS model by



(a) I = 100 (b) I = 200 (c) I = 500

(d) I = 1000 (e) I = 5000

Fig. 5. Boxplot of the performance on the DARPA Communicator data using discriminative training with different N and I.

TABLE 5
Performance comparison of MLE versus discriminative training.

Measurement	DARPA Communicator		ATIS		GENIA	
	MLE	Discriminative	MLE	Discriminative	MLE	Discriminative
Recall	87.81%	91.49%	90.21%	90.81%	61.78%	64.59%
Precision	88.13%	91.87%	92.04%	92.96%	61.16%	61.51%
F-measure	87.97%	91.68%	91.11%	91.87%	61.47%	63.01%

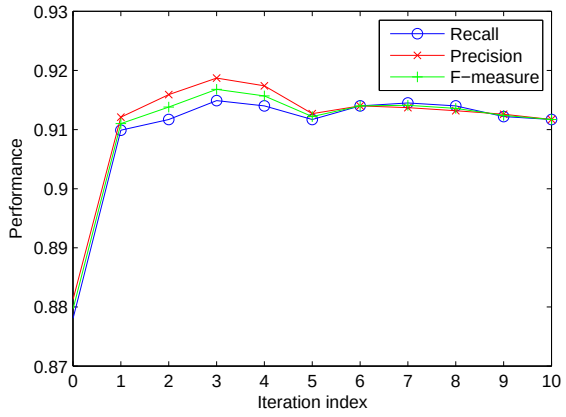
4.2%, while on the ATIS data, when $N = 5$ and $I = 100$ discriminative training achieves F-measure of 91.87%. For the more complex task on the GENIA corpus, discriminative training improves on the MLE by 2.5% where N and I are set to 5 and 200 respectively.

Figure 7 shows the performance of the HVS model versus the training iterations on the three corpora. It can be observed that discriminative training can quickly achieve the best performance on the HVS model. The best performance on the DARPA Communicator data, the ATIS data and the GENIA corpus is achieved at iteration 3, 4 and 1 respectively.

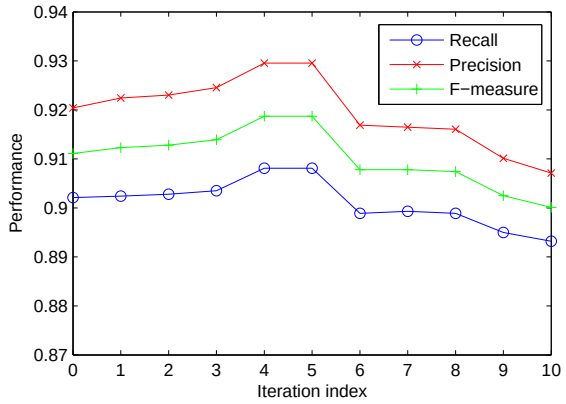
5.4 Semantic Parsing based on the Results of Speech Recognizer Output

The aforementioned experiments conducted in the travel domain used the reference transcriptions derived from the speech utterances as inputs to the semantic parser. That is, it was assumed that the speech recognizer gives 0% word error rate. Since the air travel data was originally derived from speech, a more interesting comparison would be conducted by performing semantic parsing based on the results of speech recognizer output. As we don't have the access to the DARPA Communicator speech data, experiments were only conducted on the ATIS corpus.

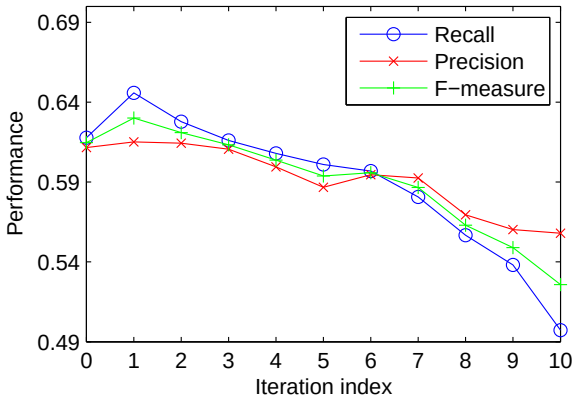
For the ATIS corpus, the training dataset consists of 4978 utterances as mentioned in section 4.2. The ATIS-3 DEC94 test set was used as our test set. The word error rate given by the speech recognizer built from the HTK toolkit [34] is 2.7%. Table 6 shows the results using MLE and discriminative training when performing semantic parsing directly on the speech recognizer output. The discriminatively trained HVS model outperforms the ML trained model by 12%.



(a) DARPA Communicator ($N = 30, I = 1000$).



(b) ATIS ($N = 5, I = 100$).



(c) GENIA corpus ($N = 5, I = 200$).

Fig. 7. Performance of the discriminatively trained HVS model vs training iterations.

TABLE 6
Performance comparison of MLE vs discriminative training on the ATIS speech recognizer output.

	MLE	Discriminative	Relative error reduction
Recall	89.46%	90.38%	8.7%
Precision	91.27%	92.67%	16.0%
F-measure	90.35%	91.51%	12.0%

5.5 Discussion

Comparing the experimental results in the travel domain on the DARPA Communicator data and the ATIS data, we found that the discriminative training approach gives the relative improvement measured in F-measure 4.2% on the DARPA Communicator data. However, the relative improvement is only 0.8% on the ATIS data. One possible reason is that the ATIS data are relatively simple while the DARPA Communicator data are more complex. Thus, MLE achieves better performance on ATIS than on the DAPRA Communicator data. As a consequence, the possible range of improvement would be smaller for ATIS. Incorporating discriminative training gives the similar performance on both corpora with the F-measure value of 91.78% obtained from ATIS and 91.68% obtained from the DARPA Communicator data.

Comparing the performance of discriminative training in the travel domain and the bioinformatics domain, the discriminative training approach achieves a relative 2.5% improvement on the GENIA corpus comparing to the 4.2% improvement on the DARPA Communicator Data. A main reason leading to the above result is that the F-measure metric was used in different ways to evaluate the model performance. In the GENIA corpus, F-measure was used to evaluate the performance of protein-protein interactions extraction. To correctly extract a protein-protein interaction, two protein names, one protein interaction keyword, and the hierarchical relations among these three terms must all be identified correctly and simultaneously. That would be only considered as one correct entry in F-measure calculation. Thus the relative improvement in F-measure in the GENIA corpus is not directly comparable to the improvement in the DARPA Communicator data.

When viewing the experimental results as relative reductions in error rate, we found that a reduction of about 30.9% and 9% were achieved in the DARPA Communicator data and the ATIS data respectively. However, for the GENIA corpus, the relative error reduction rate is only 4%. It is therefore important to test the significance levels of the performance improvement. For this purpose, we conducted the statistical test on the three corpora.

For all the three corpora, we constructed 10 models based on the different training data using discriminative training and evaluated their perfor-

TABLE 7
Statistical test on the three corpora, where p denotes the probability of the result, assuming the NULL hypothesis.

Datasets	Hypothesis H_0	t value	p value	Conclusion
DARPA Communicator Data	F-measure$_{DIS}$ = F-measure$_{MLE}(87.97\%)$ (One sample T test)	14.81	$6.3 * 10^{-8}$	$t > t_{1-0.005}(9) = 3.69$, reject H_0 at the 0.5% signification level.
ATIS Data	F-measure$_{DIS}$ = F-measure$_{MLE}(91.1\%)$ (One sample T test)	4.27	0.002	$t > t_{1-0.005}(9) = 3.69$, reject H_0 at the 0.5% signification level
GENIA corpus	F-measure$_{DIS}$ = F-measure$_{MLE}(61.47\%)$ (One sample T test)	4.18	0.002	$t > t_{1-0.005}(9) = 3.69$, reject H_0 at the 0.5% signification level
	F-measure$_{DIS}$ − F-measure$_{MLE} \geq 1.54\%$ (Paired T test)	0.04	0.484	$t < t_{1-0.005}(9) = 3.69$, can not reject $\bar{H}_0$ at the 0.5% signification level

mance on their corresponding test datasets. T-test was employed for the significance test. Table 7 lists the t values for each experiments. The probabilities of the results, assuming the NULL hypothesis, are also shown in the table. To further compare the statistical difference between the performance of MLE and the one of discriminative training on the GENIA corpus, we constructed two HVS models using MLE and discriminative training respectively, and evaluated the two models on ten different test datasets. Paired Student's T-test was used. Table 7 shows the significance test results. It can be observed that the reduction error rate of 4% between MLE and discriminated training for the GENIA corpus is indeed statistically significant.

6 Conclusion and Future Work

This paper has described how to apply discriminative training to the HVS model on the two different domains: semantic parsing on the DARPA Communicator data and the ATIS data in the travel domain and protein-protein interactions extraction on the GENIA corpus in the bioinformatics domain. The objective function is based on minimum parse error criterion. The GDP algorithm is used for estimating the parameters. Experimental results show that the proposed approach exhibits the following advantages:

- Fast convergence rate. It can achieve the best performance using only small amount of training data and converge within 3 iterations.
- Improved performance. It achieves modest improvement comparing to the ML training of the HVS model.

In future work, we plan to apply other objective functions to discriminatively train the HVS model. Also, instead of using the N-best parse results, we will explore applying discriminative training on the parse lattices directly.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments.

Appendix
Derivations of the Update formulae for Discriminative Training

To calculate the gradient of the loss function$\nabla \ell(W_i, \lambda)$, we break it into two parts and it becomes

$$\nabla \ell = \frac{\partial \ell_i}{\partial d_i} \frac{\partial d(W_i, \lambda)}{\partial \lambda} \tag{21}$$

Computing the two parts separately, we get

$$\frac{\partial \ell_i}{\partial d_i} = \frac{\gamma}{1 + e^{-\gamma d_i}} \frac{e^{-\gamma d_i}}{1 + e^{-\gamma d_i}} = \gamma \ell(d_i)(1 - \ell(d_i)) \tag{22}$$

$$\frac{\partial d(W_i, \lambda)}{\partial \lambda} = -\frac{1}{P(W_i, C_j, \lambda)} \frac{\partial P(W_i, C_j, \lambda)}{\partial \lambda} + \frac{\sum_{i, i \neq j} P(W_i, C_i, \lambda)^{\eta - 1} \frac{\partial P(W_i, C_i, \lambda)}{\partial \lambda}}{\sum_{i, i \neq j} P(W_i, C_i, \lambda)^{\eta}} \tag{23}$$

We can take the partial derivatives with respect to each of the log probability parameters. Let $\lambda =$

$\log P(n|\mathbf{c}')$, we get

$$\begin{aligned}\frac{\partial P(W_i,C_i,\lambda)}{\partial \log P(n|\mathbf{c}')} &= \partial\Big\{P(n|\mathbf{c}')^{I(C_i,n,\mathbf{c}')}\prod_{t=1,n_t\neq n\&\&\mathbf{c}_{t-1}\neq\mathbf{c}'}^{T} \\ &\quad P(n_t|\mathbf{c}_{t-1})P(c_t[1]|c_t[2..D_t])P(w_t|\mathbf{c}_t)\Big\} \\ &\quad / \ \partial \log P(n|\mathbf{c}') \\ &= I(C_i,n,\mathbf{c}')P(W_i,C_i,\lambda) \qquad (24)\end{aligned}$$

where $I(C_i,n,\mathbf{c}')$ denotes the number of times of the operation popping up n semantic tags at the current vector state $\mathbf{c}'$ in the C_i parse tree.

Thus,

$$\begin{aligned}\frac{\partial d(W_i,\lambda)}{\partial \log P(n|\mathbf{c}')} &= -\frac{1}{P(W_i,C_j,\lambda)}I(C_j,n,\mathbf{c}')P(W_i,C_i,\lambda) \\ &\quad +\frac{\sum_{i,i\neq j}P(W_i,C_i,\lambda)^\eta I(C_i,n,\mathbf{c}')}{\sum_{i,i\neq j}P(W_i,C_i,\lambda)^\eta} \\ &= -I(C_j,n,\mathbf{c}')+ \\ &\quad \sum_{i,i\neq j}I(C_i,n,\mathbf{c}')\frac{P(W_i,C_i,\lambda)^\eta}{\sum_{i,i\neq j}P(W_i,C_i,\lambda)^\eta}\end{aligned}$$

where C_j is the known correct parse tree.

In a similar way, we can get

$$\frac{\partial P(W_i,C_i,\lambda)}{\partial \log P(c[1]|c[2..D])} = I(C_i,c[1],c[2..D])P(W_i,C_i,\lambda) \qquad (25)$$

$$\frac{\partial P(W_i,C_i,\lambda)}{\partial \log P(w|\mathbf{c})} = I(C_i,w,\mathbf{c})P(W_i,C_i,\lambda) \qquad (26)$$

where $I(C_i,c[1],c[2..D])$ denotes the number of times the operation of pushing the semantic tag $c[1]$ at the current vector state $c[2..D]$ in the C_i parse tree and $I(C_i,w,\mathbf{c})$ denotes the number of times of emitting the w at the state $\mathbf{c}$ in the parse tree C_i.

And finally,

$$\begin{aligned}\frac{\partial d(W_i,\lambda)}{\partial \log P(c[1]|c[2..D])} &= -I(C_j,c[1],c[2..D])+ \\ &\quad \sum_{i,i\neq j}I(C_i,c[1],c[2..D])\frac{P(W_i,C_i,\lambda)^\eta}{\sum_{i,i\neq j}P(W_i,C_i,\lambda)^\eta}\end{aligned} \qquad (27)$$

$$\begin{aligned}\frac{\partial d(W_i,\lambda)}{\partial \log P(w|\mathbf{c})} &= -I(C_j,w,\mathbf{c})+ \\ &\quad \sum_{i,i\neq j}I(C_i,w,\mathbf{c})\frac{P(W_i,C_i,\lambda)^\eta}{\sum_{i,i\neq j}P(W_i,C_i,\lambda)^\eta}\end{aligned} \qquad (28)$$

Based on the above deductions, we can get the update formulae.

REFERENCES

[1] J. Dowding, R. Moore, F. Andry, and D. Moran, "Interleaving syntax and semantics in an efficient bottom-up parser," in *Proc. of the 32th Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, USA, 1994, pp. 110–116.

[2] W. Ward and S. Issar, "Recent improvements in the cmu spoken language understanding system," in *Proc. of the workshop on Human Language Technology*, Plainsboro, New Jerey, USA, 1994, pp. 213–216.

[3] M. Collins, "Head-driven statistical models for natural language parsing," Ph.D. dissertation, University of Pennsylvania, Philadelphia, PA, 1999.

[4] E. Charniak, "A maximum entropy inspired parser," in *1st Meeting of North American Chapter of Association for Computational Linguistics*, Seattle, Washington, 2000, pp. 132–139.

[5] Y. Normandin and S. D. Morgera, "An improved mmie training algorithm for speaker-independent, small vocabulary, continuous speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '91*, 1991, pp. 537 – 540.

[6] B. Juang, W. Chou, and C. Lee, "Statistical and discriminative methods for speech recognition," in *Speech Recognition and Understanding*, ser. NATO ASI Series, Rubio, Ed. Berlin: Springer-Verlag, 1993.

[7] W. Chou, C. Lee, and B. Juang, "Minimum error rate training based on n-best string models," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '93*, vol. 2, April 1993, pp. 652 – 655.

[8] J. Chen and F. Soong, "An n-best candidates-based discriminative training for speech recognition applications," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 206 – 216, 1994.

[9] B. Juang, W. Hou, and C. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 257 – 265, 1997.

[10] R. Pieraccini, E. Tzoukermann, Z. Gorelov, E. Levin, C. H. Lee, and J.-L. Gauvain., "Progress report on the chronus system: Atis benchmark results," in *Proc. of DARPA Speech and Natural Language Workshop*, 1992, pp. 67–71.

[11] S. Miller, R. Bobrow, R. Ingria, and R. Schwartz, "Hidden understanding models of natural language," in *Proc. of the 32th Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, June 1994, pp. 25–32.

[12] S. Miller, R. Bobrow, and R. Ingria, "statistical language processing using hidden understanding models," in *Proc. of the ARPA Human Language Technology Workshop*, Plainsboro, NJ, Mar. 1994, pp. 278–282.

[13] S. Miller, M. Bates, R. Bobrow, R. Ingria, J. Makhoul, and R. Schwartz, "Recent progress in hidden understanding models," in *Proc. of the DARPA Speech and Natural Language Workshop*, Austin, TX, Jan. 1995, pp. 276–280.

[14] R. Schwartz, S. Miller, D. Stallard, and J. Makhoul, "Language understanding using hidden understanding models," in *Proc. of Intl. Conf. on Spoken Language Processing*, Philadelphia, PA, Oct 1996.

[15] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: Analysis and applications," *Machine Learning*, vol. 32, pp. 41–62, 1998.

[16] K. Murphy and M. Paskin, "Linear time inference in hierarchical hmms," in *Proc. of Neural Information Processing Systems*, Vancouver, Canada, Dec. 2001.

[17] E. Charniak, "Immediate-head parsing for language models," in *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, 2001, pp. 124 – 131.

[18] J. Henderson, "Inducing history representations for broad coverage statistical parsing," in *Proc. of the joint meeting of the North American Chapter of the Association for Computational Linguistics and the Human Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May 2003.

[19] C. Chelba and M. Mahajan, "Information extraction using the structured language model," in *Empirical Methods in Natural Language Processing*, 2001.

[20] Y. He and S. Young, "Semantic processing using the hidden vector state model," *Computer Speech and Language*, vol. 19, no. 1, pp. 85–106, 2005.

[21] H.-K. Kuo, E. Fosle-Lussier, H. Jiang, and C. Lee, "Discriminative training of language models for speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '02*, vol. 1, April 2002, pp. 325 – 328.

[22] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86*, 1986, pp. 49–52.

[23] P. Brown, "The acoustic-modelling problem in automatic speech recognition," Ph.D. dissertation, Carnegie-Mellon University, 1987.

[24] P. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 107 – 113, 1991.

[25] V. Valtchev, J. Odell, P. Woodland, and S. Young, "Lattice-based discriminative training for large vocabulary speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '96*, vol. 2, May 1996, pp. 605 – 608.

[26] D. Povey and P. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '02*, vol. 1, April 2002, pp. 105 – 108.

[27] J. Henderson, "Discriminative training of a neural network statistical parser," in *Proc. of the 42th Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 2004, pp. 95–102.

[28] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *Annual Mathematics Statistics*, vol. 41, pp. 164–171, 1970.

[29] D. Klein and C. D. Manning, "Conditional structure versus conditional estimation in nlp models," in *Proc. the ACL-02 conference on Empirical methods in natural language processing*, University of Pennsylvania, PA, 2002, pp. 9–16.

[30] CUData, "DARPA communicator travel data. university of colorado at boulder." Avaiable from http://communicator.colorado.edu/phoenix, 2004.

[31] D. A. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and E. Shriberg, "Expanding the scope of the atis task: the atis-3 corpus," in *HLT '94: Proceedings of the workshop on Human Language Technology*. Morristown, NJ, USA: Association for Computational Linguistics, 1994, pp. 43–48.

[32] D. Zhou, Y. He, and C. K. Kwoh, "Extracting Protein-Protein Interactions from the Literature using the Hidden Vector State Model," in *International Workshop on Bioinformatics Research and Applications (LNCS 3992)*. Springer Berlin / Heidelberg, 2006, pp. 718–725.

[33] J. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "GENIA corpus–semantically annotated corpus for bio-textmining," *Bioinformatics*, vol. 19, no. Suppl 1, pp. i180–2, 2003.

[34] HTK, "Hidden Markov Model Toolkit (HTK) 3.2," Cambridge University Engineering Department, Available from http://htk.eng.cam.ac.uk, 2002.

PLACE PHOTO HERE

Deyu Zhou is currently a PhD candidate in the Informatics Research Centre at the University of Reading, UK. His interests are statistical methods for mining knowledge from texts and biomedical data mining. He received BS Degree in Mathematics in 2000 and ME Degree in Computer Science in 2003 respectively, both from Nanjing University, China.

Yulan He is a Lecturer in the Informatics Research Centre, the School of Business, the University of Reading, UK. She obtained her BASc (1st class Honors) and MEng in 1997 and 2001 respectively, both from Nanyang Technological University, Singapore. In 2004, she received her PhD degree from Cambridge University Engineering Department, UK. Between 2004 and 2007, she was an Assistant Professor with the School of Computer Engineering at Nanyang Technological University, Singapore. Her current research interests include text and data mining, machine learning, information extraction, natural langauge processing, and spoken dialogue systems.